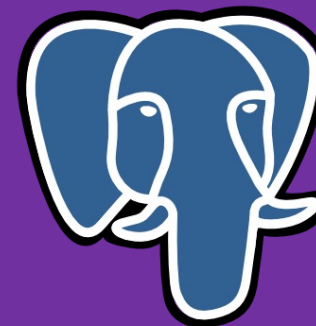




LABORATÓRIO DE BANCO DE DADOS

SQL - Consultas



PostgreSQL

- SQL Básico
- Consulta com operadores
- Consultas com agrupamento e agregações
- Subconsultas
- Consultas com condicionais
- Operações a partir de conjuntos
- Outras funções
- Common Table Expressions (CTEs)



Common Table Expressions (CTEs)

- **O que são CTEs?**

- CTEs (*Common Table Expressions*) são uma forma de criar consultas nomeadas temporárias.
- São úteis para dividir consultas complexas em partes menores e mais compreensíveis.
- Tornam o código SQL mais legível e modular.

- **Benefícios das CTEs**

- Melhoram a legibilidade do código.
- Facilitam a depuração de consultas.
- Permitem a reutilização de partes de consultas em diferentes partes do SQL.





Common Table Expressions (CTEs)

- **Sintaxe Básica de CTEs**

```
WITH nome_da_cte [(coluna1, coluna2, ...)] AS (  
    SELECT coluna1, coluna2, ...  
    FROM tabela  
    [WHERE condição]  
)
```

- **Onde:**

- nome_da_cte: Nome atribuído à CTE para referenciá-la posteriormente.
- (coluna1, coluna2, ...): Opcional, especifica as colunas retornadas pela CTE.
- SELECT ...: Consulta que define os dados na CTE.
- FROM tabela: Especifica a tabela da qual os dados são extraídos.
- WHERE condição: Opcional, define quais linhas da tabela são incluídas na CTE.





Common Table Expressions (CTEs)

- **Uso de CTEs em Consultas Principais**

- Após definir uma CTE, você pode usá-la em uma consulta principal referenciando-a pelo nome.

```
-- Liste o terceiro maior salário por departamento
WITH classificacao_salario AS (
    SELECT
        dnr,
        salario,
        DENSE_RANK() OVER (PARTITION BY dnr ORDER BY salario DESC)
        classificacao
    FROM funcionario
), SELECT * FROM classificacao_salario WHERE classificacao=3;
```





Common Table Expressions (CTEs)

- **Uso de CTEs em Consultas Principais**

- Após definir uma CTE, você pode usá-la em uma consulta principal referenciando-a pelo nome.

```
-- Recupere os três funcionários mais bem pagos em cada departamento
WITH rank_funcionario_por_departamento AS (
SELECT
d.dnome AS nome_departamento
,f.pnome AS nome_funcionario
,f.salario
,ROW_NUMBER() OVER (PARTITION BY d.dnumero ORDER BY f.salario DESC) AS classificacao
FROM departamento d JOIN funcionario f ON d.dnumero = f.dnr
) SELECT nome_departamento, nome_funcionario, salario
FROM rank_funcionario_por_departamento
WHERE classificacao <= 3;
```

